

## User friendly Help and Error map in CICS Mainframes

Taruna Yadav, Soumya Nayak, Sneha Mahajan, Anuja Munjewar, Kunal Jain

([taruna.yadav@tcs.com](mailto:taruna.yadav@tcs.com), [soumya.snayak@tcs.com](mailto:soumya.snayak@tcs.com), [sneha.mahajan@tcs.com](mailto:sneha.mahajan@tcs.com)  
[anuja.munjewar@tcs.com](mailto:anuja.munjewar@tcs.com), [jain.kunal2@tcs.com](mailto:jain.kunal2@tcs.com))

Tata Consultancy Services, India.

### ABSTRACT

The research is about giving the user a friendly interface to login into the system. The application will inform the user about the format of username and password on the login screen. It will also let the user to know about the errors that he/she may face during login.

### I. 1. INTRODUCTION

#### 1.1 Login page

The login page consists of two fields username and password and a link through which the user can view the help map to view the format of the username and password. The same link is used to view the error map when the user enters invalid data into the fields.

### 2. REQUIREMENT OF FRIENDLY INTERFACE

In today's scenario, a user comes to know about a company through its website where logging into the system is the first step. More friendly is the login interface, more the company reaches out to its customers and adds to its popularity.

### 3. TECHNICAL ASPECTS

The application is developed for Mainframe environment using COBOL for programming and CICS for the user interface. The concepts used for developing the application are:

#### 3.1 Reading dynamic cursor position

To know whether the user has placed the cursor on the help link, the position of cursor has to be known. Its position is dynamically read using BMS (Basic Mapping Support) library function "EIBCPOSN". For every terminal control

(or BMS) input operation associated with a display device, the screen cursor address (position) is placed in the EIBCPOSN field in the EIB (Execute Interface Block).

The cursor address is calculated using the formula:

$$n = (r-1) * 80 + c$$

n - absolute position of cursor

r - row number on screen

c - column number on screen

(screen resolution : 24x80)

e.g. *IF (EIBCPOSN = 1234)*

*PERFORM HELP-PARA ELSE*

*PERFORM ERROR-CHECK.*

"1234" represents the absolute position of cursor on the map.

#### 3.2 Map-on-map

In this concept, on one screen more than one maps are defined but they are not visible everytime. A map can

be made visible depending on the requirements such as if AIDkey is pressed on some link on the map then another map appears on the screen. This concept has been utilised in our application where on the login page, there is a hidden map for showing the help menu and errors. When the user presses enter key on the help link, a help map is dynamically shown on the login page (Fig 3.2 (a)).

Similarly, the user can view the errors depending upon the values that he/she enters in the username and password fields. (Fig 3.2(b)).

To make the map visible, the attributes of the map fields are dynamically changed using the copy book DFHBMSCA .

*MOVE DFHBMCRY TO USERA*

*MOVE DFHRED TO USERC*

DFHBMCRY is used to make the *USER* field attribute(A) bright.

DFHRED is used to change the color(C) of the *USER* field to red.

Example of map-on-map :

#### Implementation of random captcha

(Fig 3.2(c))

- Random captcha can be used in an application to enhance its security e.g. to ensure that registration on the website is done only by a human.
- The captcha is generated using current system time and strings which are fetched from the database.
- After fetching , unstring operation is used to display the captcha on the map fields.
- The captcha changes every two minutes.
- Aidkey is pressed to make the map visible or invisible.

## 4. ALGORITHM

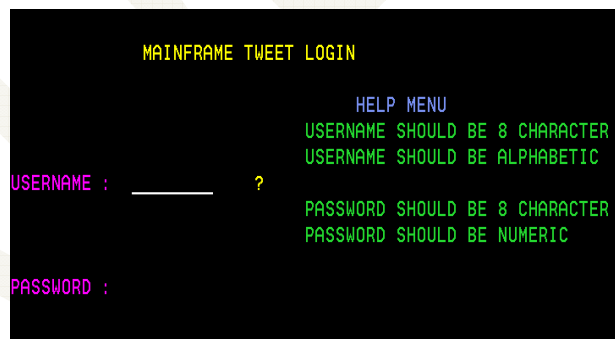
The steps followed in implementing the application are:

### 4.1 Help menu

- (a) First the login page appears containing the username and password fields .
- (b) To view the help menu, enter key is pressed on the '?' present on the login page.
- (c) To exit from the help map, enter key is pressed again.

### 4.2 Error menu

- (a) After the user enters the required fields, validations are done on the entered data .
- (b) If the data is invalid , error list can be seen by pressing enter key on the '?' .



- (c) To exit from the error map, enter key is pressed again.

( For code refer :Fig 4 )

## 5.FIGURES

Fig 3.2 (a) : Help map

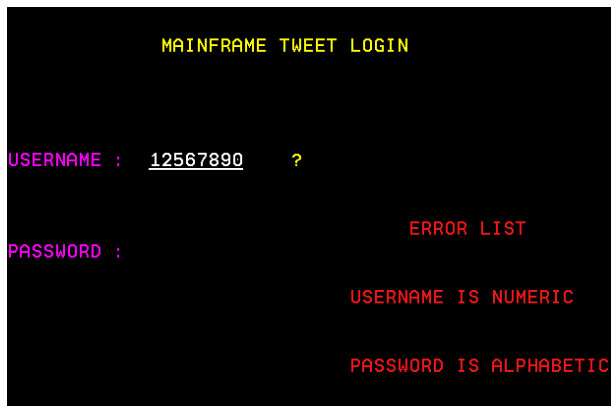
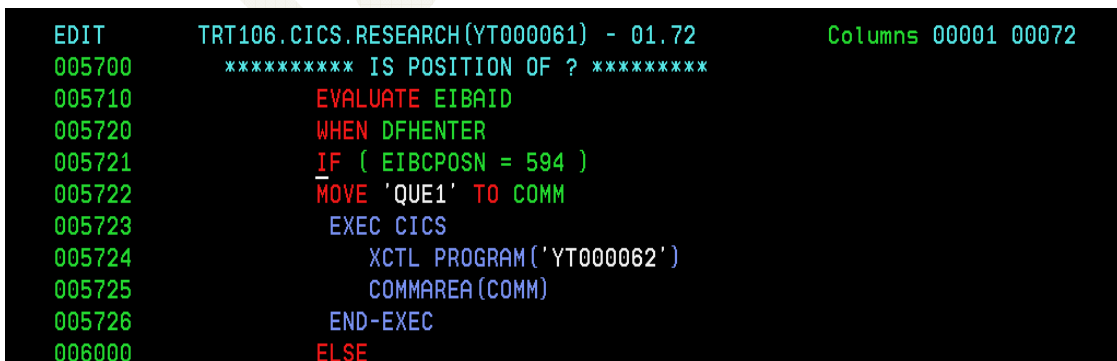


Fig 3.2(b) : Error map

Fig 3.2(c) : Implementation of random captcha

Fig 4: Screenshot of COBOL code



```
EDIT      TRT106.CICS.RESEARCH(YT000062) - 01.84
001710      EXEC CICS
001711          XCTL PROGRAM('YT000061')
001712          COMMAREA(COMM)
001713      END-EXEC
001720      END-IF.
002600      A001-SEND-MAP-PARA.
002611          IF WS-USER1 IS EQUAL TO SPACES
002612              MOVE DFHMBRY TO ERRORA
002613              MOVE DFHMBRY TO ERR1A
002614              MOVE DFHRED TO USERC
002615              MOVE DFHRED TO QC
002616              MOVE DFHRED TO USC
002617              MOVE DFHREVR TO USH
002619              PERFORM SEND-MAP
002620          ELSE IF WS-USER1 IS NUMERIC
002621              MOVE DFHRED TO USERC
```

Fig 4.2 : screenshot of code to dynamically change the attribute

## 6. CONCLUSION

For a new user, an application should have an interactive help interface through which he/she gets to know how to use the application. The interactive error map also facilitates the user to track the errors and to correct them. This application minimizes the user's efforts to search for help and therefore can be implemented in various applications such as registration systems. This application can also be used to implement a drop-down menu. The attitude and intention of a customer towards an application are affected by the ease of use, usefulness, and enjoyment thus by implementing these factors in an application it increases the customer satisfaction and makes him a repeat customer.

## 7. REFERENCES

### Books:

- [1] Introduction to the new mainframe: Z/OS Basics (An IBM red books publication).
- [2] Murarch's CICS for COBOL Programmer.
- [3] Designing and programming CICS application.
- [4] Continuous guidance by our Technical lead Mr. Robin Tommy, Tata Consultancy Services.